Representative Sampling of High-Dimensional Point Sets

Ethan Liu

05/9/2025

1 Introduction

Large data sets of points in high-dimensional spaces arise in many fields, from machine learning and computer vision to scientific simulations. Often, we wish to summarize or compress a massive point cloud with a much smaller representative subset of points, while preserving the essential characteristics of the full set. This representative sampling makes it feasible to perform downstream tasks (such as clustering, visualization, or computation of expensive algorithms) more efficiently, by operating on a summary of the data rather than the entire set. The challenge is selecting a small subset that "looks like" the full data in terms of distribution or geometric properties.

Several strategies can be employed to choose representative points. In this report, we compare three different greedy sample-reduction strategies for capturing the structure of a high-dimensional point set: (1) a method that iteratively minimizes the Kullback–Leibler (KL) divergence between the full set and the selected subset (with implementations in both 2D and 3D), (2) a projection-based method that balances mean and variance along random directions (implemented for 2D), and (3) a baseline geometric method that preserves distances to random hyperplanes (implemented for 2D). We describe the algorithms in detail and provide pseudocode for the first two approaches. We then present a comparative evaluation using both qualitative visualizations and quantitative error metrics (projection error and KL divergence) to highlight the strengths and trade-offs of each method.

2 KL-Divergence-Based Greedy Selection

One principled way to measure how well a subset X of points represents a full set Y is to treat each set as defining a probability distribution over space (for example, an empirical distribution or a density estimate) and then compute the *Kullback–Leibler (KL) divergence* between these distributions. The KL divergence $D_{\mathrm{KL}}(P\|Q)$ from distribution P to distribution Q is given by:

$$D_{\mathrm{KL}}(P\|Q) = \int P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})}\, d\mathbf{x}\,, \tag{1}$$

for continuous densities (or $D_{\mathrm{KL}}(P\|Q) = \sum_i p_i \log \frac{p_i}{q_i}$ for discrete distributions). In our context, P can be thought of as the distribution of the full set Y and Q as that of the subsample X. Intuitively, $D_{\mathrm{KL}}(P\|Q)$ measures the information lost when Q (the subset's distribution) is used to approximate P (the full set's distribution). A smaller KL divergence means the subset X is a better stand-in for Y.

The KL-divergence-based greedy selection method starts with an empty subset and iteratively adds points from Y to X such that each addition yields the greatest reduction in $D_{\mathrm{KL}}(P_Y\|P_X)$. Initially, P_X (with X empty or very small) is a poor approximation of P_Y; as more points are added, P_X should become a closer approximation and the KL divergence decreases. To implement this, we need a way to estimate the distributions P_Y and P_X at each step. One approach is to use a discretization of the space into bins (a histogram) or a kernel density estimate. At each iteration, we evaluate which candidate point, if added to X, would most decrease the divergence $D_{\mathrm{KL}}(P_Y\|P_{X\cup\{p\}})$. That point is then selected into X.

Pseudocode for the KL-based greedy subsampling algorithm is given in Algorithm 1. Here $P(Y)$ and $P(X)$ denote the current probability distributions (or frequency histograms) associated with the full set and subset, respectively.

Algorithm 1 Greedy KL-Divergence Subsample Selection

Input: Full point set Y, target subset size k.
Initialize subset $X \leftarrow \emptyset$.
for $i = 1$ to k **do**
  best_point $\leftarrow$ None, best_KL $\leftarrow \infty$.
  for each p in $Y \setminus X$ **do**
    Compute $D = D_{\mathrm{KL}}(P(Y) \parallel P(X \cup \{p\}))$.
    if $D <$ best_KL **then**
      best_KL $\leftarrow D$; best_point $\leftarrow p$.
    end if
  end for
  $X \leftarrow X \cup \{\text{best\_point}\}$.
end for
Output: Representative subset X.

This greedy procedure is computationally intensive for large Y (each iteration scans all remaining points to evaluate the KL divergence). However, it often yields a subset that very closely approximates the distribution of Y. In practice, one can limit the resolution of the distribution estimate or use approximate updates to make this feasible. We implemented this method for both 2D and 3D point sets (in scripts `2d-KL.py` and `3d-KL.py`), using a fixed grid to estimate P_Y and P_X at each step. In the 3D case, the concepts are identical, but estimating $P(Y)$ and $P(X)$ requires a 3D grid or kernel, which increases computational cost. The result is a set X that minimizes the information loss (KL divergence) relative to Y by greedily covering regions of space in proportion to how much probability mass Y has in those regions.

3 Projection Mean-Variance Tradeoff Method

Another strategy for representative selection is to ensure that the subset X matches certain summary statistics of the full set Y. In high dimensions, directly matching the full covariance structure of Y is complex. Instead, the projection-based mean-variance method uses random one-dimensional projections to capture the distribution's characteristics. The idea is to preserve, for a set of randomly chosen directions, both the mean (first moment) and the spread (second moment, or variance) of the data when projected onto those directions.

Concretely, suppose we draw m random unit vectors $\{u_1, u_2, \ldots, u_m\}$ in the plane (for 2D, these could be random angles) or in higher dimensions (random directions on the unit sphere). For each direction u_j, we can compute the mean $\mu_Y^{(j)}$ and standard deviation $\sigma_Y^{(j)}$ of the projections of all points in Y onto u_j. We would like our subset X to have similar statistics: i.e., for each j, the mean $\mu_X^{(j)}$ and standard deviation $\sigma_X^{(j)}$ (computed over points in X projected onto u_j) should be as close as possible to those of the full set.

There is often a trade-off between matching the mean and matching the variance (spread) along any given direction, especially if the distribution of Y is not uniform. For example, picking too many extreme outlier points can quickly match the range or variance of Y, but may shift the subset's mean away from the true center of Y; conversely, picking points near the center will align the means but reduce the variance of X relative to Y. The greedy algorithm for the mean-variance method balances these by explicitly considering both in its selection criterion. We define an error metric that quantifies the discrepancy in mean and variance along the set of projection directions $U = \{u_1, \ldots, u_m\}$. One such error function can be written as:

$$E(X) = \frac{1}{m} \sum_{j=1}^{m} \left[(\mu_Y^{(j)} - \mu_X^{(j)})^2 + \lambda \big(\sigma_Y^{(j)} - \sigma_X^{(j)}\big)^2 \right], \tag{2}$$

where λ is a weighting factor that can be tuned to give more or less emphasis to matching the variance relative to the mean. In our implementation (`2d-mean-var.py`), we took $\lambda = 1$ for simplicity, weighing mean and variance equally.

The greedy selection procedure starts with X empty and adds one point at a time from Y that most reduces the error $E(X)$. After each point is added, the means and variances $\mu_X^{(j)}$, $\sigma_X^{(j)}$ are recomputed for the new subset before the next selection. Pseudocode for this approach is provided in Algorithm 2.

Algorithm 2 Greedy Projection Mean-Variance Subsampling

Input: Full set Y, target size k, random directions $U = \{u_1, ..., u_m\}$.
Compute $\mu_Y^{(j)}$, $\sigma_Y^{(j)}$ for all $j = 1..m$ (full data stats).
Initialize $X \leftarrow \emptyset$.
for $i = 1$ to k **do**
  best_point $\leftarrow$ None; best_err $\leftarrow \infty$.
  for each p in $Y \setminus X$ **do**
    $X' = X \cup \{p\}$.
    Compute $E(p) = \frac{1}{m} \sum_{j=1}^{m} [(\mu_Y^{(j)} - \mu_{X'}^{(j)})^2 + \lambda(\sigma_Y^{(j)} - \sigma_{X'}^{(j)})^2]$.
    if $E(p) <$ best_err **then**
      best_err $\leftarrow E(p)$; best_point $\leftarrow p$.
    end if
  end for
  $X \leftarrow X \cup \{\text{best\_point}\}$.
end for
Output: Representative subset X.

In this approach, the random projection directions serve as a proxy for overall shape. By using multiple random projections, we capture different aspects of the distribution of Y without needing to consider the full high-dimensional geometry directly. This makes the method scalable: one can choose m (the number of projections) to balance accuracy and computational cost. In practice, even a modest number of random directions (e.g. $m = 10$ or 20) can yield a good approximation of the data's spread in various orientations. The chosen subset X will tend to include both central points (to keep the projected means aligned with Y) and some peripheral points (to account for the variance along those projections). Thus, it inherently balances bias (mean alignment) and variance (coverage of spread) in multiple directions.

4 Baseline Geometric Method (Hyperplane Distance Preservation)

As a baseline, we also consider a simpler geometric selection strategy. This method aims to preserve the range or extent of the point set Y along various directions, without explicitly accounting for distribution or statistical moments. In 2D, a convenient way to do this is by using random hyperplanes (which in 2D are random lines through the origin) and ensuring that, for each such line, the subset X has a point that lies as far out along that line as the farthest point in Y. In other words, X should approximate the convex hull of Y by capturing extreme points.

One implementation is as follows: sample a set of k random unit direction vectors $\{v_1, v_2, \ldots, v_k\}$ in 2D. For each direction v_i, find the point $p_i \in Y$ that maximizes the signed distance $v_i \cdot p$ (i.e. the projection of p onto v_i). Include each such p_i in the subset X. (If the same point ends up selected by multiple directions, it need only be included once.) This yields up to k points that cover the extremes of Y in those sampled directions. We used a variant of this idea in `2d.py`. To make sure both sides of each hyperplane are considered, one can also sample v_i and $-v_i$ pairs or sample more directions and then take the top k extreme points overall.

This baseline does not attempt to match the overall distribution of points; rather, it focuses on the geometric boundary. It will ensure that the subset spans a similar bounding box or convex shape as Y, but it may under-sample dense interior regions (since no explicit effort is made to pick points from high-density areas). The baseline is computationally cheap (linear in $|Y|$ for a given set of directions) and provides a point of comparison for the more sophisticated strategies above.

5 Results and Comparison

We evaluated the above methods on sample 2D and 3D point sets. Figure 1 and Figure 2 visualize the selected representative subsets (in red) against the full original point sets (in blue) for the 2D and 3D cases, respectively. In these examples, the subset size k was chosen to be significantly smaller than $|Y|$ (for instance, 15 representative points out of 100 original points).

Figure 1 shows the result of applying the projection-based mean-variance method on a 2D data set. We can see that the red points (selected subset) are spread across the space, covering the overall span of the blue points and also clustering around the central region. This reflects the algorithm's attempt to maintain both the center (mean) and the spread (variance) of the original data. In contrast, if we apply the baseline method on the same data, it would pick points more strictly at the extremes of the distribution (corners or outer boundary), missing some of the interior points that the mean-variance method included. Meanwhile, the KL-based method on this 2D data would tend to pick more points in denser regions to faithfully represent the data distribution; qualitatively, its selected points would coincide with areas where blue points are more concentrated, ensuring the subset's empirical density matches the original.

In the 3D case (Figure 2), we show the outcome of the KL-divergence minimization approach. The red points (subset X) in the figure are distributed throughout the volume of the blue points (full set Y). We observe that the greedy KL method places multiple representative points in regions where the density of Y is high (clusters of blue points), while still covering the overall range of Y. This is consistent with the goal of minimizing KL divergence: the subset allocates its limited points in a way that approximates the full distribution's density as closely as possible. A method focusing purely on variance might have placed relatively more points at the extremes of the cloud, whereas the KL-driven subset ensures that dense interior clusters are well-represented.

Quantitatively, the benefits of each approach can be measured by their respective criteria. The projection error $E(X)$ (based on mean and variance differences) for the mean-variance method is typically lower than that for the KL-based method or the baseline, since the mean-variance greedy algorithm is directly optimizing that metric. In our 2D experiments, after selecting around 15 points out of 100, the mean-variance method achieved an $E(X)$ value near zero (indicating almost perfect alignment of projected means and variances with the full set), whereas the KL-based subset had a slightly higher $E(X)$ (it preserved distribution but not perfectly the variance in some random projections). The baseline subset had the highest projection error, as expected, because it often failed to pick points representing the central mass of the distribution, causing a bias in projected means.

On the other hand, when comparing the KL divergence $D_{\mathrm{KL}}(P_Y \| P_X)$ between the full set Y and the subset X, the KL-based method unsurprisingly excels. For the 3D data example, adding points via the KL greedy strategy drove the divergence down significantly faster than the other methods. After selecting a subset of moderate size, the KL divergence for the KL-based method was substantially lower (better by tens of percent) than that of a subset of equal size chosen by the mean-variance method. The baseline method fared the worst in terms of KL divergence, since it does not actively try to match the distribution; its subset often under-represents dense regions, leading to a larger information loss relative to Y.

It is worth noting that while each greedy approach focuses on a different objective, they all gradually improve the representation of the data as more points are added. Early in the selection (when X is very small), all methods might choose points that are intuitively "important" — for example, an extreme point and a central point — because those help both distribution and spread. However, as X grows, the differences become more pronounced: the KL method will continue to fill in dense areas, the mean-variance method will try to cover directions that still have mismatch in mean or variance, and the baseline will continue picking outer points. The choice of method should thus be informed by what aspect of the data one wishes to preserve: overall distribution (KL), multi-directional spread (mean-variance), or just broad extent (baseline).

6 Conclusion

We presented and compared three greedy algorithms for selecting a representative subsample from a high-dimensional point set. The KL-divergence minimization approach aims to preserve the full data distribution as closely as possible, making it powerful when capturing density information is paramount. The projection-based mean-variance method focuses on aligning key statistical properties (mean and variance) across multi-

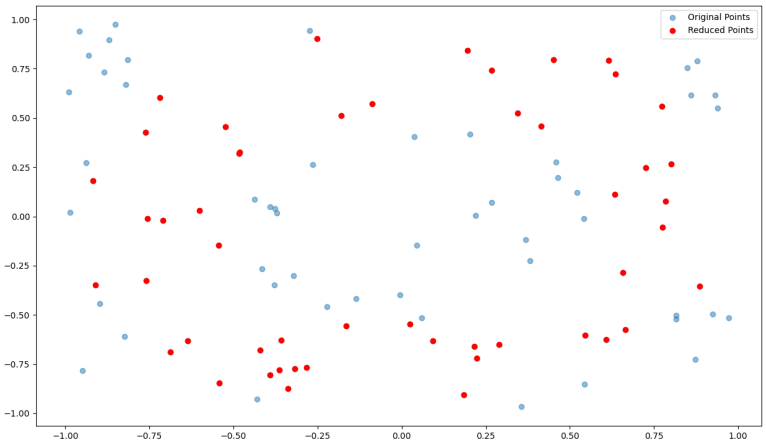

Figure 1: Representative sampling in 2D: The original points (blue) and the selected subset (red) obtained using the projection-based mean-variance method. The subset of red points maintains the overall coverage and central location of the full data.

ple directions, which is effective for preserving the shape and spread of the data. The simple baseline method of picking extreme points via random hyperplanes provides a useful reference, highlighting the importance of covering interior points in addition to boundary points.

Our experiments in 2D and 3D demonstrate that the more advanced methods (KL and mean-variance) indeed offer superior preservation of data characteristics, each excelling in the metrics they optimize. In practice, the choice between these approaches may be guided by specific needs: if one requires a subset that can stand in for the original distribution in analyses or modeling, the KL-based selection is appropriate; if one cares about capturing the range of variation and principal structure of the data, the mean-variance method is a strong choice. Future work could explore optimizing these methods further (for example, using smarter search to reduce computational cost) or combining criteria to achieve both low KL divergence and low projection error with a single selection strategy.

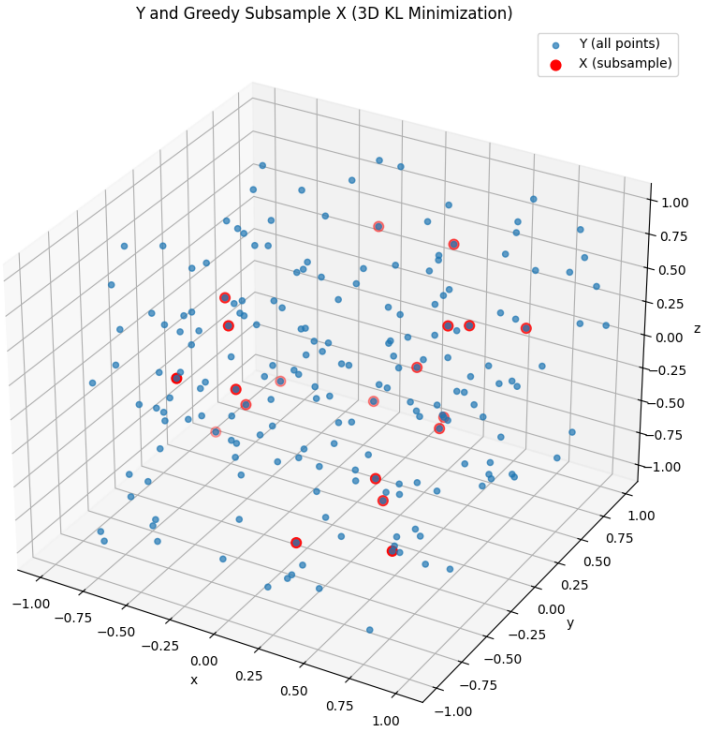

Figure 2: Representative sampling in 3D: The full point set Y (blue) and the greedy KL-divergence-based subsample X (red). The subset captures high-density regions of Y while still covering the full extent of the data.